# WHEN DOES FREQUENCY DECOMPOSITION BENEFIT PHYSICS-INFORMED NEURAL NETWORKS? A PRELIMINARY ABLATION STUDY

A PREPRINT

**Shubham Rai** *
shubham.rai@bibha.ai
**bibha.ai**

August 14, 2026

**ABSTRACT**

Partial differential equations (PDEs) often have high-frequency and multi-scale features that neural networks struggle to approximate. Physics-Informed Neural Networks (PINNs) build the governing equations directly into training, but suffer from *spectral bias*: they learn low-frequency components faster than high-frequency ones. Techniques such as Fourier feature embeddings and sinusoidal activations address this, but most studies assume they help across the board without checking which spectral regimes actually benefit. We introduce a dual-branch, spectrally-gated architecture (DBSG-PINN) that splits low- and high-frequency components into separate subnetworks joined by an adaptive gate, and use it to run a partially controlled ablation of frequency decomposition and spectral routing. We test this on five one-dimensional benchmark PDEs, ranging from smooth, single-scale problems to oscillatory, multi-scale ones. Frequency decomposition helps most on the spectrally complex benchmarks, cutting relative $L_2$ error by up to $59.2\%$ on a multimodal wave problem, but gives little benefit on smoother PDEs. On one benchmark (1D Wave), it performs substantially worse than a simpler fixed-combination variant. The gate's benefit scales with how spectrally rich the target solution is: the full model's advantage over the ablations is largest on multi-scale benchmarks and smallest (or negative) on single-scale ones, consistent with the gate exploiting frequency structure rather than acting as noise, though we do not directly visualize or quantify its spatial activations in this study. All results come from a single training seed across five 1D benchmarks, so we present this as an exploratory study meant to raise questions rather than answer them, and outline the additional seeds and benchmarks needed to test whether the pattern holds.

***Keywords*** Physics Informed Neural Networks · Spectral Bias · Frequency Decomposition · Ablation Study · Partial Differential Equations

## 1 Introduction

Partial differential equations (PDEs) are the mathematical backbone for modelling physical, biological, and engineering systems, including fluid flow, heat transfer, wave propagation, diffusion, reaction kinetics, and electromagnetic phenomena. Most of these equations come from conservation laws and first principles, yet closed-form solutions exist only for a narrow class of idealized cases. Once nonlinear dynamics, complex geometries, heterogeneous materials, or high-dimensional parameter spaces enter the picture, an analytical solution is usually out of reach. That is why numerical methods – finite differences, finite elements, spectral solvers – carry most of the load in practice. They work, but not cheaply: mesh generation, repeated simulation runs, and heavy compute are the price, and that price gets steeper for inverse problems, uncertainty quantification, and parameter estimation.

*Corresponding author.

Scientific Machine Learning (SciML) takes a different route by folding the governing physical laws directly into a data-driven model, rather than relying on observed data alone – the network is pushed toward physically consistent solutions without needing as much labelled data. Physics-Informed Neural Networks (PINNs) (Raissi et al., 2019) are probably the most influential framework to come out of this line of work for forward and inverse PDE problems: governing equations, boundary conditions, and initial conditions all get folded into the training objective through automatic differentiation, so whatever the network learns has to satisfy the underlying physics. PINNs have since shown up in computational fluid dynamics, inverse modelling, biomedical engineering, and geophysical simulation, and are more or less a default tool in SciML at this point (Karniadakis et al., 2021; Cuomo et al., 2022; Luo et al., 2025).

That said, conventional PINNs are not without problems: optimization pathologies, imbalance between competing loss terms, and poor convergence on stiff or highly nonlinear PDEs have all been reported (Krishnapriyan et al., 2021; Wang et al., 2021, 2022). *Spectral bias* is probably the most studied of these – networks trained with gradient descent pick up low-frequency components of a target function well before high-frequency ones. Rahaman et al. (2019) first showed this in deep networks, and the Neural Tangent Kernel (NTK) later gave a reason why: low-frequency functions line up with the dominant kernel eigenmodes and so converge faster (Jacot et al., 2018). In practice this means PINNs often struggle with localized discontinuities, sharp gradients, oscillatory behaviour, or several interacting spatial or temporal scales – wave propagation (Moseley et al., 2020), reaction–diffusion systems, and other high-frequency or multi-scale PDEs are where this tends to bite hardest (Mustajab et al., 2024).

A number of frequency-aware fixes have been proposed for spectral bias, and they roughly split into four camps. One enriches the input itself: Fourier feature embeddings help a network approximate high-frequency functions (Tancik et al., 2020), and the periodic activations in Sinusoidal Representation Networks (SIREN) do something similar for oscillatory signals and their derivatives (Sitzmann et al., 2020). A second camp works on the optimization process directly, through adaptive activation functions, adaptive loss balancing, and learning-rate annealing (Jagtap et al., 2020a; Wang et al., 2021). A third tackles scalability with domain decomposition – Conservative PINNs (cPINNs), Extended PINNs (XPINNs), and Finite Basis PINNs (FBPINNs) all split the domain into subregions to make optimization and local accuracy more tractable (Jagtap et al., 2020b; Jagtap and Karniadakis, 2020; Moseley et al., 2023). And a fourth leans on gating mechanisms or deeper residual architectures for scalability and stability (Stiller et al., 2020; He et al., 2016). Between them, these advances have pushed the range of problems PINNs can handle considerably further.

Despite this progress, most existing work focuses on building more sophisticated architectures to fight spectral bias, and pays less attention to *when* these mechanisms actually help. Most frequency-aware methods assume that richer spectral representations are broadly useful across PDE types. But physical systems differ a lot in their spectral character: some PDEs are smooth and low-frequency, while others involve localized oscillations, several interacting frequency modes, or complex multi-scale dynamics. This raises a natural question: does explicit frequency decomposition consistently improve accuracy, or does its value depend on the spectral character of the PDE? As far as we know, few studies have directly compared frequency-decomposed architectures against matched-capacity ablations across PDEs of varying spectral complexity. That gap motivates the exploratory study we present here.

To study this question, we build the **Dual-Branch Spectral-Gated Physics-Informed Neural Network (DBSG-PINN)**, a controlled experimental framework for examining the role of frequency decomposition in PINNs. It consists of two subnetworks that each specialize in a different frequency regime: a low-frequency branch using hyperbolic tangent activations to capture smooth solution components, and a high-frequency branch using sinusoidal activations to model oscillatory behaviour. Their outputs are combined by an adaptive gate that learns spatially varying mixing weights, letting the model balance the two representations according to local solution behaviour. Each component can be removed without changing the optimization procedure or (roughly) the network's capacity, which is what makes ablation studies possible: we can isolate the individual contributions of frequency decomposition and adaptive spectral routing.

We test the framework on five benchmark PDEs spanning a range of spectral complexity: the Burgers equation, Reaction–Diffusion equation, Allen–Cahn equation, Wave equation, and a Multimodal Wave equation, ranging from smooth, single-scale solutions to oscillatory, multi-scale dynamics. For each benchmark, we compare the full DBSG-PINN against three ablation variants, made by removing the high-frequency branch, the low-frequency branch, or the adaptive gate on its own, while keeping network capacity and optimization settings matched as closely as the architecture allows. This is meant to reduce, though not remove, the risk that any performance difference comes from model complexity or training strategy rather than frequency decomposition itself; we discuss what this control does and does not achieve in Section 4.

Across the five benchmarks, frequency decomposition clearly pays off on the spectrally complex ones: up to 59.2% lower relative $L_2$ error on Multimodal Wave, plus better spectral recovery there more broadly. On the smoother, single-scale problems the benefit shrinks or disappears – a simpler fixed-combination variant edges out the full gated model on Burgers, and beats it outright on 1D Wave. We want to be careful about what this does and does not show:

these are observations from one seed across five benchmarks, not a general claim about frequency-decomposed PINNs as a class.

Beyond accuracy, we also looked at what the learned gate contributes across benchmarks. Its benefit is largest on the benchmarks with the richest spectral content and smallest (or negative) on the simplest ones – a pattern consistent with the gate exploiting frequency structure rather than acting as noise, though we infer this from benchmark-level comparisons rather than from a direct visualization of the gate's spatial activations. That said, this comes from one trained model per benchmark, and it needs independent verification – ideally including an explicit visualization of $g(x,t)$ against local spectral content – before we can call it a general property of the architecture.

**Our Contributions**

- We build DBSG-PINN, a dual-branch architecture combining low- and high-frequency subnetworks through an adaptive spectral gate.
- We design an ablation framework that removes the low-frequency branch, high-frequency branch, or gate individually, keeping training settings matched exactly and per-branch capacity matched only approximately – depth and width are not identical between the low- and high-frequency branches on most benchmarks (Section 2.3).
- We report an initial comparison across five 1D benchmark PDEs of varying spectral complexity – meant to surface a pattern worth investigating further, not to establish a general rule.
- We show a preliminary, benchmark-level pattern in which the gate's contribution scales with the spectral complexity of the target solution, though we do not directly visualize or quantify its spatial activations here.

## 2 Methodology

### 2.1 Physics-Informed Neural Networks

PINNs belong to the broader class of Scientific Machine Learning (SciML) models that fold the governing physical laws directly into the training objective rather than relying only on labelled data (Raissi et al., 2019). Concretely, this means minimizing the residual of the governing PDE alongside the initial and boundary conditions, pushing the network toward solutions consistent with the underlying physics rather than merely fitting data points.

Consider a general nonlinear PDE

$$\frac{\partial u(\mathbf{x},t)}{\partial t} + \mathcal{N}[u(\mathbf{x},t)] = 0, \tag{1}$$

where $u(\mathbf{x},t)$ is the unknown solution, $\mathbf{x}$ is the spatial coordinate, and $\mathcal{N}(\cdot)$ is a nonlinear differential operator.

The solution is approximated by a neural network

$$u(\mathbf{x},t) \approx u_\theta(\mathbf{x},t), \tag{2}$$

where $\theta$ are the trainable network parameters.

Using automatic differentiation, the PDE residual is computed as

$$r_\theta(\mathbf{x},t) = \frac{\partial u_\theta(\mathbf{x},t)}{\partial t} + \mathcal{N}[u_\theta(\mathbf{x},t)]. \tag{3}$$

All the spatial and temporal derivatives needed to evaluate $r_\theta(\mathbf{x},t)$ come from automatic differentiation (Baydin et al., 2018), which computes exact derivatives through the network graph instead of relying on finite-difference approximations.

The overall PINN objective is a weighted sum of the governing-equation residual and the initial and boundary condition losses,

$$\mathcal{L} = \lambda_r \mathcal{L}_{PDE} + \lambda_{IC}\mathcal{L}_{IC} + \lambda_{BC}\mathcal{L}_{BC}, \tag{4}$$

The scalar weights $\lambda_r$, $\lambda_{IC}$, and $\lambda_{BC}$ balance these terms. Poorly chosen weights are a known cause of the gradient imbalance and stiffness problems reported in PINN training, which is why prior work has explored adaptive or self-weighting schemes (McClenny and Braga-Neto, 2020; Wang et al., 2022). Here we fix the weights per benchmark (Table 6) so that all ablation variants share the same loss landscape; we leave adaptive weighting for future work.

where

$$\mathcal{L}_{PDE} = \frac{1}{N_r} \sum_{i=1}^{N_r} \left| r_\theta(\mathbf{x}_i, t_i) \right|^2, \tag{5}$$

and

$$\mathcal{L}_{IC} = \frac{1}{N_{IC}} \sum_{i=1}^{N_{IC}} \left| u_\theta(\mathbf{x}_i, 0) - u_0(\mathbf{x}_i) \right|^2, \tag{6}$$

Equation 1 is written as a first-order-in-time PDE for generality, in which case the value term above is the only initial condition needed. For the second-order-in-time (wave-type) benchmarks in Section 3 (Multimodal Wave and 1D Wave), the governing equation also specifies an initial velocity $u_t(\mathbf{x}, 0) = v_0(\mathbf{x})$ alongside the initial value $u(\mathbf{x}, 0) = u_0(\mathbf{x})$; we enforce both terms in $\mathcal{L}_{IC}$,

$$\mathcal{L}_{IC} = \frac{1}{N_{IC}} \sum_{i=1}^{N_{IC}} \left[ \left| u_\theta(\mathbf{x}_i, 0) - u_0(\mathbf{x}_i) \right|^2 + \left| \partial_t u_\theta(\mathbf{x}_i, 0) - v_0(\mathbf{x}_i) \right|^2 \right], \tag{7}$$

with $\partial_t u_\theta(\mathbf{x}_i, 0)$ again obtained via automatic differentiation and both terms weighted equally within $\mathcal{L}_{IC}$. For the first-order-in-time benchmarks (Burgers, Allen–Cahn, Reaction–Diffusion), only the value term applies, as in the equation above.

$$\mathcal{L}_{BC} = \frac{1}{N_{BC}} \sum_{i=1}^{N_{BC}} \left| u_\theta(\mathbf{x}_i, t_i) - g(\mathbf{x}_i, t_i) \right|^2. \tag{8}$$

Following common practice in the PINN literature, we minimize $\mathcal{L}$ in two stages: a first-order Adam optimizer (Kingma and Ba, 2015) handles rapid initial descent, then a second-order L-BFGS optimizer (Liu and Nocedal, 1989) refines convergence near the minimum. Table 6 reports the iteration budgets used for each benchmark.

### 2.2 Proposed DBSG-PINN Architecture

The Dual-Branch Spectral-Gated Physics-Informed Neural Network (DBSG-PINN) splits the approximation of low- and high-frequency solution components into two subnetworks, combined by a learned gate. The input coordinates $(x, t)$ go in parallel to all three subnetworks — the low-frequency branch, the high-frequency branch, and the gate network — which are trained jointly, end to end, by backpropagating the composite loss (Eq. 4) through the combined output. No branch is pretrained or frozen on its own.

The low-frequency branch is represented as

$$u_{\mathrm{lo}} = f_{\mathrm{lo}}(\mathbf{x}, t; \theta_{\mathrm{lo}}), \tag{9}$$

where $f_{\mathrm{lo}}(\cdot)$ uses hyperbolic tangent activations — a standard smooth nonlinearity backed by classical universal approximation results for feedforward networks (Hornik et al., 1989) — to model smooth solution components.

Similarly, the high-frequency branch is defined as

$$u_{\mathrm{hi}} = f_{\mathrm{hi}}(\mathbf{x}, t; \theta_{\mathrm{hi}}), \tag{10}$$

where $f_{\mathrm{hi}}(\cdot)$ uses sinusoidal activations to better represent oscillatory and multi-scale features, following the periodic-activation design of SIREN-style networks (Sitzmann et al., 2020).

We use "low-frequency" and "high-frequency branch" as shorthand for each branch's intended inductive bias – smooth $\tanh$ nonlinearities versus oscillatory $\sin(\omega x)$ activations at a fixed frequency $\omega$ – not as a formal spectral decomposition.

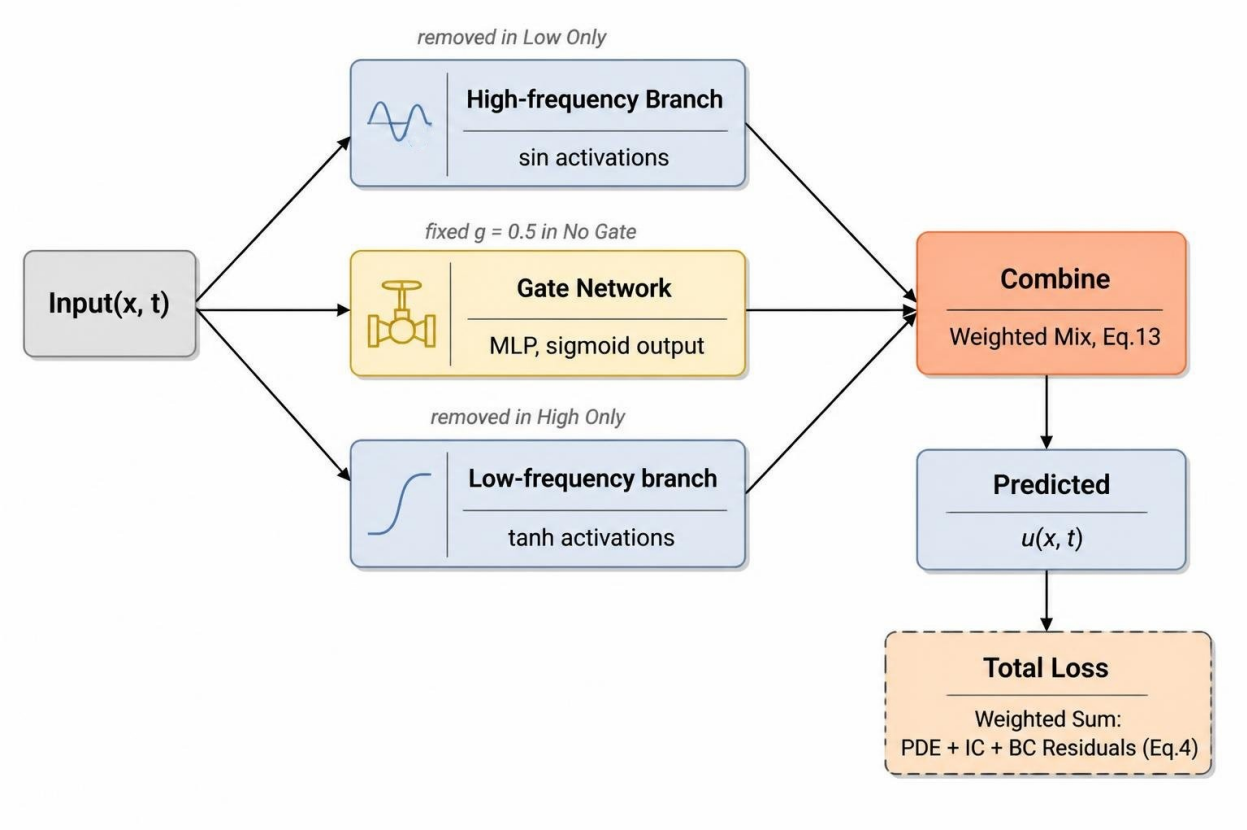



Figure 1: DBSG-PINN architecture

Nothing in the architecture constrains $u_{\text{lo}}$ to contain only low-frequency content or $u_{\text{hi}}$ to contain only high-frequency content: either branch is, in principle, free to represent any function its activation function and depth allow, and the gate is what determines how much each contributes at a given $(x, t)$. Throughout this paper we use "frequency decomposition" to mean this soft, activation-driven inductive bias, not a guarantee of exact spectral separation.

To combine the two representations, a lightweight gating network predicts an adaptive weighting coefficient, in the spirit of mixture-of-experts architectures that route between subnetworks with a trained gate (Jacobs et al., 1991; Shazeer et al., 2017). Those architectures usually gate more than two experts and so use a softmax; since DBSG-PINN mixes exactly two branches, we use a scalar sigmoid gate instead,

$$g(\mathbf{x}, t) = \sigma\left(f_g(\mathbf{x}, t; \theta_g)\right), \tag{11}$$

where $\sigma(\cdot)$ is the sigmoid activation function, so that

$$0 \leq g(\mathbf{x}, t) \leq 1. \tag{12}$$

The final prediction is a convex combination of the two branches,

$$u_\theta(\mathbf{x}, t) = g(\mathbf{x}, t) u_{\text{hi}}(\mathbf{x}, t) + \left(1 - g(\mathbf{x}, t)\right) u_{\text{lo}}(\mathbf{x}, t). \tag{13}$$

The PDE residual for this architecture is therefore

$$r_\theta(\mathbf{x}, t) = \frac{\partial u_\theta}{\partial t} + \mathcal{N}[u_\theta], \tag{14}$$

where, again, all derivatives come from automatic differentiation. Figure 1 shows the overall framework.

### 2.3 Ablation Variants

Each ablation variant is trained separately, from a fresh random initialization, on its own copy of the computational graph above — built by structurally removing or fixing the relevant component (the high-frequency branch, the low-frequency branch, or the gate), not by defining a different architecture. Every variant uses the same optimizer, iteration budget, collocation strategy, and loss weights as the full model for that benchmark (Table 6); only the active architectural components change. This reduces the risk that any performance difference comes from optimizer settings or training procedure rather than frequency decomposition itself. It does not, on its own, control for seed variance, which we discuss as a limitation in Section 4.

**The low- and high-frequency branches are not strictly capacity-matched.** We state this up front because it bears directly on how LowOnly and HighOnly should be read: depth and width differ between the two branches on most benchmarks (details below), so a LowOnly-vs-HighOnly comparison is not a clean like-for-like test of the two activation functions alone – some of the difference could come from capacity rather than frequency bias.

- **Full (DBSG-PINN):** both branches active, combined via the learned gate $g(x,t)$ as in Eq. 13.
- **NoGate:** both branches active, but $g(x,t)$ in Eq. 13 is fixed at $0.5$ instead of learned, giving a simple average of the two branches.
- **LowOnly:** the high-frequency branch is dropped ($u_\theta = u_{\mathrm{lo}}$); the gate and high-frequency branch are not used at all.
- **HighOnly:** the low-frequency branch is dropped in the same way ($u_\theta = u_{\mathrm{hi}}$).

For each PDE, the low- and high-frequency branches use a roughly similar number of hidden layers and parameters (Table 6), but depth and width are not identical between the two branches: on most benchmarks the high-frequency branch is wider (e.g. 64 vs. 32 units on Multimodal Wave, 128 vs. 64 on 1D Wave), and depth differs by one or two layers on Burgers, Reaction–Diffusion, and Allen–Cahn. We chose these settings empirically per benchmark rather than by a strict matching rule, so LowOnly and HighOnly are not exactly capacity-matched ablations — they are only roughly comparable in scale, with activation function ($\tanh$ vs. $\sin(\omega x)$) as the main intended difference. Even when depth and width do match, that does not fully equate capacity in a functional sense: $\tanh$ and $\sin(\omega x)$ networks of the same size can still differ in expressivity, so any parameter matching here is a partial control, not a complete one. With that caveat in mind, the comparison is meant to highlight the inductive bias from the activation function rather than gross differences in network size, though the per-benchmark width and depth choices remain an uncontrolled source of variation. Together, the four variants probe two questions: does explicit frequency separation with a learned gate help (Full vs. NoGate), and is either single-frequency regime alone enough for the target solution (LowOnly / HighOnly vs. Full)?

### 2.4 Evaluation Metrics

We report the following metrics throughout Section 3, evaluated on a uniform $100 \times 100$ ($N_x \times N_t$) grid over the problem domain (Table 6 gives the collocation spacing used during training; the evaluation grid is separate from, and coarser-to-finer than, the training resolution depending on benchmark). Let $u_\theta$ be the trained network's prediction and $u_{\mathrm{exact}}$ the reference solution, both evaluated on this grid. For Multimodal Wave, Reaction–Diffusion, and 1D Wave, $u_{\mathrm{exact}}$ is the closed-form analytical solution given in Section 3. Allen–Cahn and Burgers admit no simple closed form on this domain, so for these two benchmarks $u_{\mathrm{exact}}$ denotes a high-resolution numerical reference solution rather than an analytical one; we say more about how each is obtained in the corresponding subsection of Section 3.

**Relative $L_2$ error.** Following the evaluation convention Raissi et al. (2019) introduced for PINNs (used there on, among others, the Burgers and Allen–Cahn benchmarks studied here too) and adopted as a default reporting metric in the DeepXDE library (Lu et al., 2021), the global relative $L_2$ error over the evaluation grid is

$$\mathrm{Rel.L}_2 = \frac{\left(\sum_{i,j} \left(u_\theta(x_i,t_j) - u_{\mathrm{exact}}(x_i,t_j)\right)^2\right)^{1/2}}{\left(\sum_{i,j} u_{\mathrm{exact}}(x_i,t_j)^2\right)^{1/2}}. \tag{15}$$

**$L_\infty$ error.** Reported alongside relative $L_2$ error in the same references (Raissi et al., 2019; Lu et al., 2021), this is the pointwise maximum absolute error over the evaluation grid,

$$L_\infty = \max_{i,j} \left|u_\theta(x_i,t_j) - u_{\mathrm{exact}}(x_i,t_j)\right|. \tag{16}$$

**Mean PDE residual.** Reporting the governing-equation residual on the evaluation grid as a post-training diagnostic, alongside pointwise accuracy metrics, follows practice established for PINN software such as DeepXDE (Lu et al., 2021). Unlike the metrics above, the PDE residual reported in Tables 1–5 is not the autodiff-based residual $r_\theta$ minimized during training (Eq. 3, following Raissi et al., 2019). It is a post-hoc diagnostic: after training, we apply second-order central finite differences to the trained prediction $u_\theta$ on the evaluation grid, approximating each benchmark's own governing-equation operator. For the wave-type benchmarks (Eqs. 20 and 31), for example, this is $\hat{r}_\theta(x_i, t_j) = \hat{u}_{tt}(x_i, t_j) - c^2 \hat{u}_{xx}(x_i, t_j)$, with $\hat{u}_{xx}$ and $\hat{u}_{tt}$ evaluated by central differences at spacing $\Delta x$, $\Delta t$ over the interior evaluation grid; the other benchmarks use the equivalent finite-difference form of their own PDE (Eqs. 23, 26, 29). We report the mean absolute residual over interior grid points,

$$\overline{|R|} = \frac{1}{(N_x - 2)(N_t - 2)} \sum_{i=2}^{N_x - 1} \sum_{j=2}^{N_t - 1} |\hat{r}_\theta(x_i, t_j)| \,. \tag{17}$$

Because this is a finite-difference approximation computed separately from training, rather than the exact autodiff residual $r_\theta$, it should be read as a post-hoc consistency check on the trained solution, not as a direct measure of the quantity minimized during optimization.

**Spectral error, HF recovery, and LF recovery.** These diagnostics adapt, for a post-training evaluation setting, the Fourier-domain approach Rahaman et al. (2019) and Xu et al. (2020) used to characterize spectral bias during training: comparing the amplitude spectra of a network's output against the target function across frequency bands to show that low frequencies are learned first (*spectral bias/F-Principle*). We apply the same comparison to the trained solution instead of to training dynamics: for each time slice $t_j$, we compute the discrete Fourier amplitude spectra of the predicted and exact solutions along $x$, $\widehat{u}_\theta(k, t_j) = |\mathrm{FFT}_x[u_\theta(\cdot, t_j)](k)|$ and $\widehat{u}_{\mathrm{exact}}(k, t_j)$ defined the same way, for wavenumbers $k = 1, \ldots, K_{\max}$ (we use $K_{\max} = 20$). We then average over the $N_t$ time slices to get the mean amplitude spectra $\bar{u}_\theta(k) = \frac{1}{N_t} \sum_j \widehat{u}_\theta(k, t_j)$ and $\bar{u}_{\mathrm{exact}}(k)$, defined the same way. We exclude the zero-frequency (DC) component from all spectral metrics because it reflects the mean solution level rather than its oscillatory content. The *spectral error* is the normalized $L_2$ distance between the two mean spectra over all remaining modes,

$$\mathrm{SpecErr} = \frac{\|\bar{u}_\theta(2{:}K_{\max}) - \bar{u}_{\mathrm{exact}}(2{:}K_{\max})\|_2}{\|\bar{u}_{\mathrm{exact}}(2{:}K_{\max})\|_2}. \tag{18}$$

We split the remaining modes into a low-frequency band $\mathcal{K}_{\mathrm{LF}}$ (modes 2–5) and a high-frequency band $\mathcal{K}_{\mathrm{HF}}$ (modes 6–$K_{\max}$), and define a per-band normalized error $e_{\mathcal{K}} = \|\bar{u}_\theta(\mathcal{K}) - \bar{u}_{\mathrm{exact}}(\mathcal{K})\|_2 \,/\, \|\bar{u}_{\mathrm{exact}}(\mathcal{K})\|_2$. The *LF recovery* and *HF recovery* scores are then

$$\mathrm{Recovery}(\mathcal{K}) = \frac{1}{1 + e_{\mathcal{K}}}, \qquad \mathcal{K} \in \{\mathcal{K}_{\mathrm{LF}}, \mathcal{K}_{\mathrm{HF}}\}, \tag{19}$$

a score of 1 means the two spectra match perfectly in that band, and the score drops toward 0 as the normalized error grows. Unlike Eq. 15, these are diagnostic scores rather than accuracy metrics: they capture how well the amplitude spectrum is recovered in a given band, regardless of any phase or pointwise error elsewhere.

A caveat applies when a benchmark's reference solution has little or no energy in one of these two bands. Reaction–Diffusion's solution is a single spatial mode ($\sin(\pi x)$, Section 3) with essentially all its Fourier energy at the lowest retained mode, and 1D Wave's solution is likewise a single spatial mode ($k = 4$) sitting inside the LF band as we define it here. In both cases the reference amplitude in the opposite band is close to zero, so the per-band normalized error $e_{\mathcal{K}}$ divides by a near-zero denominator; the resulting recovery score is then driven mainly by whatever small, largely non-physical high-frequency content the trained network happens to produce (interpolation artifacts, activation-induced ripple, discretization leakage from the FFT treating a non-periodic Dirichlet solution as periodic), rather than by how well a genuine spectral feature is recovered. We still report these columns for completeness and consistency across benchmarks, but on Reaction–Diffusion (HF band) and 1D Wave (HF band, since its only real content sits in the LF band at $k = 4$) the recovery score should be read as a noise-floor diagnostic rather than a substantive accuracy measure.

## 3 Results

Every number below comes from a single training run per model per benchmark (seed in Table 6), so we read these as observations from an initial study rather than statistically confirmed effects – Section 4 says more about what that limitation should and should not be taken to mean. Metric definitions (relative $L_2$ error, spectral error, HF/LF recovery) are in Section 2.4.

### 3.1 1D Multimodal Wave Equation

We consider the multimodal wave equation

$$u_{tt} = c^2\, u_{xx}, \qquad (x,t) \in [0,1] \times [0,1], \quad c = 1, \tag{20}$$

with a multimodal initial condition composed of modes $k \in \{1, 3, 5\}$ with amplitudes $a_k \in \{1.0,\ 0.7,\ 0.4\}$,

$$u(x,0) = \sum_{k \in \{1,3,5\}} a_k \sin(k\pi x), \qquad u_t(x,0) = 0, \tag{21}$$

and Dirichlet boundary conditions $u(0,t) = u(1,t) = 0$. This admits the closed-form solution

$$u(x,t) = \sum_{k \in \{1,3,5\}} a_k \sin(k\pi x)\cos(k\pi c\, t). \tag{22}$$

Table 1: Ablation Study of DBSG-PINN on the 1D Multimodal Wave Equation (single seed)

| Model | Relative L2 Error ↓ | $L_\infty$ Error ↓ | Mean PDE Residual ↓ | Spectral Error ↓ | HF Recovery ↑ | LF Recovery ↑ |
|---|---|---|---|---|---|---|
| **DBSG-PINN (Full)** | **0.0762** | **0.1632** | 0.0218 | **0.0371** | **0.8603** | **0.9658** |
| NoGate | 0.1161 | 0.2235 | **0.0206** | 0.0840 | 0.8019 | 0.9239 |
| LowOnly | 0.1869 | 0.4649 | 0.0378 | 0.1387 | 0.7235 | 0.8798 |
| HighOnly | 0.1632 | 0.2877 | 0.0257 | 0.1380 | 0.7797 | 0.8796 |

### 3.2 1D Allen–Cahn Equation

The Allen–Cahn equation, subject to periodic boundary conditions, takes the form

$$u_t = \varepsilon\, u_{xx} + 5u - 5u^3, \qquad (x,t) \in [-1,1] \times [0,1], \tag{23}$$

with diffusion coefficient $\varepsilon = 10^{-4}$, initial condition

$$u(x,0) = x^2 \cos(\pi x), \tag{24}$$

and periodic boundary conditions

$$u(-1,t) = u(1,t), \qquad u_x(-1,t) = u_x(1,t). \tag{25}$$

This equation has no simple closed-form solution. Following the same benchmark as originally introduced for PINNs (Raissi et al., 2019), we evaluate against a high-resolution numerical reference solution (obtained independently of the network being trained) rather than an analytical one; all "exact"/reference figures and metrics for Allen–Cahn refer to this numerical reference.

Table 2: Ablation Study of DBSG-PINN on the Allen–Cahn Equation (single seed)

| Model | Relative L2 ↓ | $L_\infty$ Error ↓ | Mean PDE Residual ↓ | Spectral Error ↓ | HF Recovery ↑ | LF Recovery ↑ |
|---|---|---|---|---|---|---|
| **DBSG-PINN (Full)** | **0.1038** | **0.7416** | 0.0107 | **0.0700** | **0.7985** | **0.9459** |
| NoGate | 0.5892 | 1.8905 | 0.0183 | 0.3571 | 0.5898 | 0.7446 |
| LowOnly | 0.1108 | 0.7547 | **0.0084** | 0.0734 | 0.7933 | 0.9430 |
| HighOnly | 0.6717 | 1.8997 | 0.0519 | 0.4116 | 0.5472 | 0.7174 |

### 3.3 1D Burgers Equation

We use the viscous Burgers equation with Dirichlet boundary conditions,

$$u_t + u\, u_x = \nu\, u_{xx}, \qquad (x,t) \in [-1,1] \times [0,1], \tag{26}$$

with viscosity $\nu = 0.01/\pi$, initial condition

$$u(x,0) = -\sin(\pi x), \tag{27}$$

and boundary conditions

$$u(-1,t) = 0, \qquad u(1,t) = 0. \tag{28}$$

This viscosity and initial condition admit no simple closed-form solution either; following the same benchmark as in (Raissi et al., 2019), we evaluate against a high-resolution numerical reference solution rather than an analytical one.

Table 3: Ablation Study of DBSG-PINN on the 1D Burgers Equation (single seed)

| Model | Relative L2 ↓ | $L_\infty$ Error ↓ | Mean PDE Residual ↓ | Spectral Error ↓ | HF Recovery ↑ | LF Recovery ↑ |
|---|---|---|---|---|---|---|
| DBSG-PINN (Full) | 0.02398 | 0.16720 | **0.26810** | **0.01447** | **0.94387** | 0.99098 |
| **NoGate** | **0.02383** | **0.15984** | 0.26836 | 0.01495 | 0.94079 | **0.99107** |
| LowOnly | 0.02816 | 0.17997 | 0.27716 | 0.01593 | 0.93585 | 0.99089 |
| HighOnly | 0.02455 | 0.16313 | 0.26866 | 0.01631 | 0.93767 | 0.99081 |

### 3.4 1D Reaction–Diffusion Equation

This benchmark is governed by the linear reaction–diffusion equation

$$u_t = D\,u_{xx} + \lambda u, \qquad (x,t) \in [0,1] \times [0,1], \tag{29}$$

with diffusion coefficient $D = 0.01$ and reaction rate $\lambda = 2$, initial condition $u(x,0) = \sin(\pi x)$, and Dirichlet boundary conditions $u(0,t) = u(1,t) = 0$, with exact solution

$$u(x,t) = \sin(\pi x)\,\exp\big((\lambda - D\pi^2)\,t\big). \tag{30}$$

Unlike Multimodal Wave, this solution is a single spatial mode – a $\sin(\pi x)$ profile that grows or decays in time depending on the sign of $\lambda - D\pi^2$ – so we include this benchmark as a smooth, low-order case rather than a spectrally complex one; we return to what that implies for the ablation results in Section 4.

Table 4: Ablation Study of DBSG-PINN on the 1D Reaction–Diffusion Equation (single seed)

| Model | Relative L2 Error ↓ | $L_\infty$ Error ↓ | Mean PDE Residual ↓ | Spectral Error ↓ | HF Recovery ↑ | LF Recovery ↑ |
|---|---|---|---|---|---|---|
| **DBSG-PINN (Full)** | $\mathbf{2.24 \times 10^{-4}}$ | $\mathbf{6.30 \times 10^{-3}}$ | $\mathbf{2.30 \times 10^{-3}}$ | $\mathbf{2.63 \times 10^{-4}}$ | 0.9956 | **0.9998** |
| NoGate | $3.25 \times 10^{-4}$ | $6.95 \times 10^{-3}$ | $2.51 \times 10^{-3}$ | $3.44 \times 10^{-4}$ | 0.9980 | 0.9997 |
| LowOnly | $4.77 \times 10^{-4}$ | $6.30 \times 10^{-3}$ | $2.53 \times 10^{-3}$ | $5.37 \times 10^{-4}$ | **0.9985** | 0.9995 |
| HighOnly | $1.01 \times 10^{-3}$ | $8.85 \times 10^{-3}$ | $5.00 \times 10^{-3}$ | $1.18 \times 10^{-3}$ | 0.9937 | 0.9988 |

### 3.5 1D Wave Equation

The final benchmark is the standard 1D wave equation

$$u_{tt} = c^2\,u_{xx}, \qquad (x,t) \in [0,1] \times [0,1], \quad c = 1, \tag{31}$$

with a single-mode initial condition of wavenumber $k = 4$,

$$u(x,0) = \sin(4\pi x), \qquad u_t(x,0) = 0, \tag{32}$$

and Dirichlet boundary conditions $u(0,t) = u(1,t) = 0$, whose exact solution is

$$u(x,t) = \sin(4\pi x)\cos(4\pi c\,t). \tag{33}$$

Table 5: Ablation Study of DBSG-PINN on the 1D Wave Equation (single seed)

| Model | Relative L2 ↓ | $L_\infty$ Error ↓ | Mean PDE Residual ↓ | Spectral Error ↓ | HF Recovery ↑ | LF Recovery ↑ |
|---|---|---|---|---|---|---|
| DBSG-PINN (Full) | 0.05134 | 0.08320 | 0.02931 | 0.02699 | 0.63539 | 0.97581 |
| NoGate | **0.02397** | 0.05143 | 0.02711 | **0.00850** | 0.78424 | **0.99326** |
| LowOnly | 0.49170 | 0.67322 | **0.02240** | 0.35938 | 0.29961 | 0.73716 |
| HighOnly | 0.02514 | **0.04897** | 0.03340 | 0.01565 | **0.98891** | 0.98475 |

## 4 Discussion

Every number in this section comes from one trained model per benchmark per variant. We treat the pattern below as a hypothesis to test with more seeds and benchmarks, not as a settled result. To avoid repeating that caveat in every paragraph, we say it once here and come back to it, with specifics, in the Limitations paragraph.

**Decomposition helps most on genuinely multi-scale targets.** On Multimodal Wave – the one benchmark whose solution is an explicit superposition of multiple spatial modes ($k \in \{1, 3, 5\}$, Section 3) – the full model beats every ablation on nearly every metric (Table 1). LowOnly and HighOnly each specialize in one part of the spectrum, and each does measurably worse than the full model at recovering the content the other branch was meant to handle, which is roughly what we expected going in: separating the branches pays off when a solution genuinely mixes frequency regimes. Reaction–Diffusion also shows Full leading on every accuracy metric (Table 4) – LowOnly edges it out only on HF recovery, which Section 2.4 already flags as a noise-floor diagnostic rather than a substantive measure for this benchmark's near-single-mode spectrum – but that benchmark's solution is a single spatial mode (Section 3), not a spectrally rich target, so we do not think the same explanation applies there; we return to this discrepancy below.

**On Allen–Cahn, LowOnly nearly matches the full model.** Full still comes out ahead (0.1038 vs. 0.1108 relative $L_2$ for LowOnly), though only just, whereas the gap to NoGate and HighOnly is large (Table 2). The Allen–Cahn solution is smooth and low-frequency apart from sharp, localized transition layers, so a sinusoidal high-frequency branch mostly gets in the way here: HighOnly and the unweighted NoGate average both pay for that mismatch. It looks like the gate's real job on this benchmark is suppressing a branch that is not helping rather than blending two useful ones, though Full and LowOnly sit close enough together that a different seed could flip the ranking.

**On Burgers, gating showed no meaningful edge either way.** NoGate (fixed at $g = 0.5$) and the full model end up close together across all six metrics for Burgers (Table 3): 0.0240 vs. 0.0238 relative $L_2$, say, with each variant ahead on roughly half the columns and by less than 5% either way. Viscosity smooths Burgers out to something close to single-scale, so there is not much spectral content left for an adaptive gate to route between – a wash is more or less what we would expect here.

**On 1D Wave, the full model did meaningfully worse than NoGate.** Unlike Burgers, this gap is not small: NoGate leads the full model on every metric we tracked (Table 5), by around 53% on relative $L_2$ and 68% on spectral error. The benchmark uses a single wavenumber ($k = 4$), so we didn't expect the gate to have much to route between — instead, learning a gate actively hurt performance compared to a fixed average. We don't have a confident explanation for why the effect is this large on Wave but not Burgers; that's a priority for multi-seed follow-up.

**Interpreting the gate.** Stepping back, the gate's benefit is largest on the benchmark with the richest spectral content (Multimodal Wave), smaller but still positive on Allen–Cahn (smooth except for sharp transition layers), roughly neutral on Burgers, and negative on 1D Wave, whose target is a single spatial mode. This much fits what we would expect if the gate's aggregate benefit tracks spectral complexity across benchmarks. Reaction–Diffusion is an exception: its solution is also a single spatial mode, yet the full model still leads on every accuracy metric there (HF recovery aside, a noise-floor diagnostic for this benchmark), so whatever the gate is doing on that benchmark, it does not look like it is exploiting multi-mode spectral structure in the way our framing above assumes. We flag this as an open question rather than force it into the pattern, and note again that none of this is based on a direct visualization of $g(x, t)$ – only on comparing aggregate performance across benchmarks of differing spectral content.

**Limitations.** Three limitations bound how far these results should be read. *Seeds:* every result comes from a single training seed (Table 6), so we have no variance estimate. This matters most for Allen–Cahn's Full-vs-LowOnly gap, which is small enough to plausibly flip under a different seed, and arguably even more for the 1D Wave Full-vs-NoGate gap — that gap is large in this run, but with a single seed we can't tell whether it's a real, reproducible effect or just an unlucky training run for the full model. *Capacity matching:* branch depth and width are only roughly comparable, not identical, across the low- and high-frequency branches for most benchmarks (Section 2.3); even where sizes do match, equal parameter count doesn't guarantee equal functional capacity, since $\tanh$ and $\sin(\omega x)$ networks of the same size aren't guaranteed equal expressivity. *Scope:* all five benchmarks are one-dimensional, so we can't say whether the pattern holds in higher dimensions or under stronger nonlinearity.

## 5 Conclusion

This work asked when frequency decomposition actually helps Physics-Informed Neural Networks, instead of assuming its value is universal. We introduced DBSG-PINN, a dual-branch architecture with an adaptive spectral gate built specifically for ablation: each of its three components — the low-frequency branch, the high-frequency branch, and the gating mechanism — can be removed without changing the optimization procedure, so we can compare their individual contributions directly.

The pattern across the five benchmarks is not a single clean story. Decomposition helped most on Multimodal Wave, the one benchmark whose solution is a genuine superposition of spatial modes, cutting relative $L_2$ error by up to 59.2%.

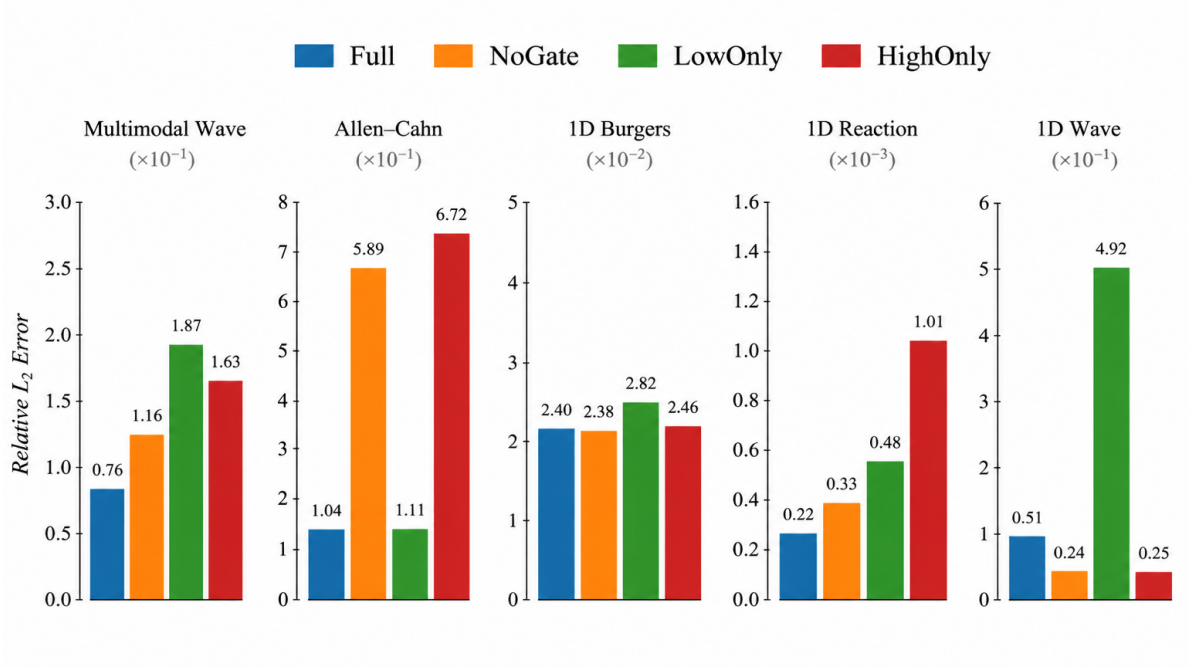



Figure 2: Relative $L_2$ error by ablation variant across the five benchmarks (single seed per bar). DBSG-PINN (Full) leads on Multimodal Wave, Allen–Cahn, and Reaction–Diffusion, while NoGate leads on Burgers and 1D Wave.

The full gated model also beat every ablation on every accuracy metric for Reaction–Diffusion (HF recovery aside, a noise-floor diagnostic there), even though that benchmark's solution is a single spatial mode rather than a spectrally rich one – we flag this as an open question rather than a confirmation of our starting hypothesis. Allen–Cahn was more mixed, with Full ahead but only barely so over LowOnly, suggesting the gate's job there was really screening out a poorly matched high-frequency branch rather than blending two useful ones. Burgers gave learned gating no meaningful edge over a simple fixed average, and 1D Wave went the other way entirely: NoGate beat the full model by a wide margin on every metric tracked, a gap large enough that it needs multi-seed confirmation before being read as more than a flag. Where the gate did help, its benefit tracked the benchmark's spectral complexity – a preliminary, interpretable signal, not a proven mechanism.

None of this settles when frequency decomposition helps, and it is not meant to. The single-seed, five-benchmark, one-dimensional setting is a real limitation, and the closer comparisons here, Allen–Cahn's Full-vs-LowOnly gap especially, should not be read past what is actually reported. What it does offer is a concrete, testable pattern: decomposition helps most on genuinely multi-scale solutions and least on smooth, single-scale ones. The natural next step is checking whether that pattern survives more seeds and a wider benchmark suite, before drawing any firmer conclusions for PINN practice.

**Future work.** The most immediate next step is repeating this ablation study across multiple seeds to get variance estimates and confirm whether the reported orderings are stable; several of the comparisons in Section 4 are close enough that this is a prerequisite for any stronger claim. Beyond that, natural extensions include: (i) extending the benchmark suite to 2D and time-dependent multi-scale PDEs (e.g., 2D Navier–Stokes, wave scattering); (ii) building a lightweight diagnostic, based on the spectral content of the initial/boundary data, that estimates in advance whether frequency decomposition is likely to help for a given PDE; and (iii) testing whether the gate network's link to spectral complexity holds up as an interpretability signal beyond the specific architecture studied here.

## Acknowledgements

Figures 1 and 2 were generated with the assistance of Paper Banana. Claude (Anthropic) was used for language refinement of the manuscript text. All AI-assisted content was reviewed and verified by the author, who takes full responsibility for the research, experiments, analysis, and conclusions presented in this work.

## References


Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. (2018). Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18(153):1–43.

Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., and Piccialli, F. (2022). Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31.

Jagtap, A. D. and Karniadakis, G. E. (2020). Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics*, 28(5):2002–2041.

Jagtap, A. D., Kawaguchi, K., and Karniadakis, G. E. (2020a). Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics*, 404:109136.

Jagtap, A. D., Kharazmi, E., and Karniadakis, G. E. (2020b). Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 365:113028.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R., and Mahoney, M. W. (2021). Characterizing possible failure modes in physics-informed neural networks. *Advances in neural information processing systems*, 34:26548–26560.

Liu, D. C. and Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528.

Lu, L., Meng, X., Mao, Z., and Karniadakis, G. E. (2021). DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228.

Luo, K., Zhao, J., Wang, Y., Li, J., Wen, J., Liang, J., Soekmadji, H., and Liao, S. (2025). Physics-informed neural networks for pde problems: A comprehensive review. *Artificial Intelligence Review*, 58(10):323.

McClenny, L. and Braga-Neto, U. (2020). Self-adaptive physics-informed neural networks using a soft attention mechanism. *arXiv preprint arXiv:2009.04544*.

Moseley, B., Markham, A., and Nissen-Meyer, T. (2020). Solving the wave equation with physics-informed deep learning. *arXiv preprint arXiv:2006.11894*.

Moseley, B., Markham, A., and Nissen-Meyer, T. (2023). Finite basis physics-informed neural networks (fbpinns): a scalable domain decomposition approach for solving differential equations. *Advances in Computational Mathematics*, 49(4):62.

Mustajab, A. H., Lyu, H., Rizvi, Z., and Wuttke, F. (2024). Physics-informed neural networks for high-frequency and multi-scale problems using transfer learning. *Applied Sciences*, 14(8):3204.

Pal, A. (2023). Lux: Explicit Parameterization of Deep Neural Networks in Julia.

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., and Courville, A. (2019). On the spectral bias of neural networks. In *International conference on machine learning*, pages 5301–5310. PMLR.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. (2020). Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473.

Stiller, P., Bethke, F., Böhme, M., Pausch, R., Torge, S., Debus, A., Vorberger, J., Bussmann, M., and Hoffmann, N. (2020). Large-scale neural solvers for partial differential equations. In *Smoky Mountains Computational Sciences and Engineering Conference*, pages 20–34. Springer.

Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., and Ng, R. (2020). Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547.

Wang, S., Teng, Y., and Perdikaris, P. (2021). Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081.

Wang, S., Yu, X., and Perdikaris, P. (2022). When and why pinns fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449:110768.

Xu, Z.-Q. J., Zhang, Y., Luo, T., Xiao, Y., and Ma, Z. (2020). Frequency principle: Fourier analysis sheds light on deep neural networks. *Communications in Computational Physics*, 28(5):1746–1767.

Zubov, K., McCarthy, Z., Ma, Y., Calisto, F., Pagliarino, V., Azeglio, S., Bottero, L., Luján, E., Sulzer, V., Bharambe, A., Vinchhi, N., Balakrishnan, K., Upadhyay, D., and Rackauckas, C. (2021). NeuralPDE: Automating physics-informed neural networks (PINNs) with error approximations. *arXiv preprint arXiv:2107.09443*.

# A Appendix

## A.1 Hyperparameter Configuration

Table 6 reports the hyperparameter configuration we used for DBSG-PINN across all five benchmark PDEs, including each branch's architecture (activation function, hidden layers, width), the gate network configuration, and training settings (collocation grid spacing, Adam/L-BFGS iteration counts, loss weights, and random seed). All ablation variants (NoGate, LowOnly, HighOnly) reuse the same per-benchmark configuration from Table 6 for their active components. Training budget and per-branch capacity therefore stay matched across variants for a given benchmark, though this alone gives no variance estimate, since only one seed was run per configuration (Section 4).

Table 6: Hyperparameter configuration of DBSG-PINN across the five benchmark PDEs. All entries use a single random seed per benchmark; see Section 4 for the resulting limitation on statistical confidence.

| Hyperparameter | Burgers | Reaction–Diff. | Allen–Cahn | Multimodal Wave | 1D Wave |
|---|---|---|---|---|---|
| *Low-frequency branch* | | | | | |
| Activation | tanh | tanh | tanh | tanh | tanh |
| Hidden layers | 3 | 3 | 4 | 3 | 3 |
| Width | 24 | 32 | 64 | 32 | 64 |
| *High-frequency branch* | | | | | |
| Activation | $\sin(\omega x)$ | $\sin(\omega x)$ | $\sin(\omega x)$ | $\sin(\omega x)$ | $\sin(\omega x)$ |
| Frequency $\omega$ | 2 | 2 | 1 | 2 | 4 |
| Hidden layers | 4 | 4 | 2 | 3 | 3 |
| Width | 32 | 32 | 128 | 64 | 128 |
| *Gate network* | | | | | |
| Activation | tanh / sigmoid | tanh / sigmoid | tanh / sigmoid | tanh / sigmoid | tanh / sigmoid |
| Hidden layers | 1 | 2 | 1 | 2 | 2 |
| Width | 16 | 16 | 32 | 16 | 32 |
| Sigmoid gain | 6 | 6 | 1 | 1 | 1 |
| *Training* | | | | | |
| Collocation grid spacing | 0.02 | 0.02 | 0.025 | 0.02 | 0.01 |
| Adam learning rate | $5 \times 10^{-4}$ | $5 \times 10^{-4}$ | $5 \times 10^{-4}$ | $5 \times 10^{-4}$ | $5 \times 10^{-4}$ |
| Adam iterations | 15,000 | 10,000 | 18,000 | 15,000 | 25,000 |
| L-BFGS iterations | 3,000 | 3,000 | 6,000 | 4,000 | 8,000 |
| IC loss weight | 1 | 1 | 50 | 1 | 1 |
| Random seed | 42 | 42 | 42 | 42 | 42 |

**Implementation.** All models are implemented in Julia using NeuralPDE.jl (Zubov et al., 2021) for the physics-informed training loop and Lux.jl version 1.2.3 (Pal, 2023) for network parameterization, with grid-based collocation (`GridTraining`) at the per-benchmark spacings listed above. Training uses Adam (Kingma and Ba, 2015) at the learning rate given above, followed by L-BFGS (Liu and Nocedal, 1989) for the iteration counts listed, both with default (Julia `Optim.jl`/`OptimizationOptimJL`) line-search settings. All evaluation metrics in Section 3 (relative $L_2$, $L_\infty$, mean PDE residual, spectral error, HF/LF recovery) are computed on a uniform $100 \times 100$ evaluation grid, as stated in Section 2.4; this grid is independent of, and in most cases finer than, the training collocation spacing above.

## A.2 1D Multimodal Wave Equation: Qualitative Comparison Across Variants

Figures 3–7 show, for each ablation variant, the exact solution, the model's prediction, the pointwise absolute error, and (where shown) the FFT-based spectral recovery on the 1D Multimodal Wave benchmark. All panels share a consistent color scale within a figure to support direct visual comparison; each figure corresponds to a single trained model (Section 4).

## A.3 1D Burgers Equation: Qualitative Comparison Across Variants

Figures 8–12 show the same reference-solution / prediction / absolute-error / spectral-recovery comparison for the 1D Burgers benchmark (reference here meaning the high-resolution numerical solution described in Section 3, not a closed-form exact solution).

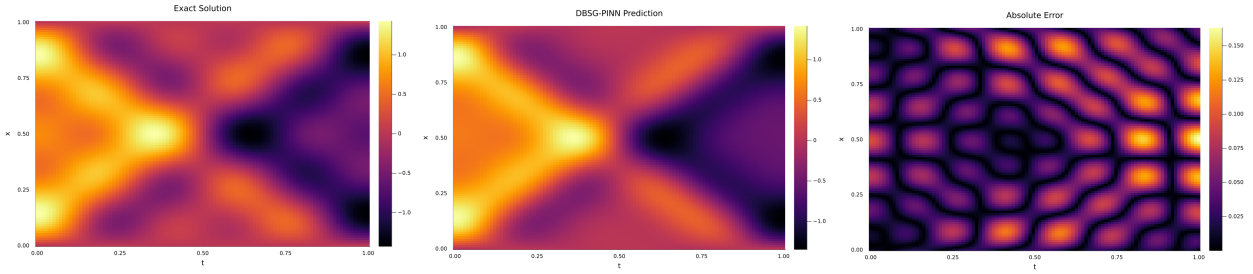



Figure 3: 1D Multimodal Wave, Full DBSG-PINN: exact solution, prediction, and absolute error.

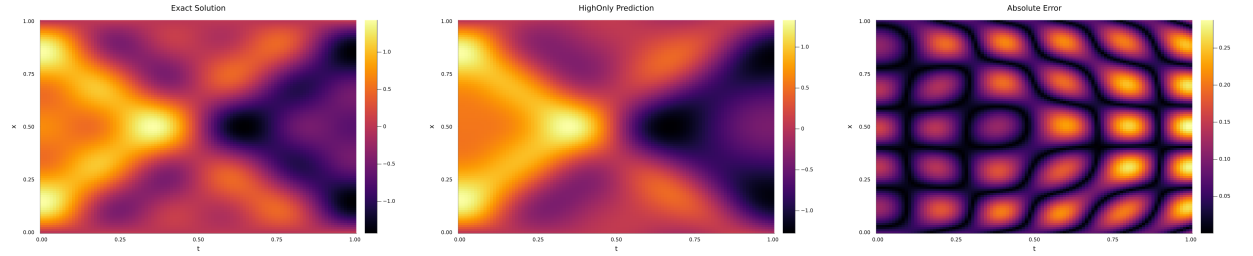



Figure 4: 1D Multimodal Wave, HighOnly variant.

### A.4 1D Allen–Cahn Equation: Qualitative Comparison Across Variants

Figures 13–17 show the same comparison for the 1D Allen–Cahn benchmark.

### A.5 1D Reaction–Diffusion Equation: Qualitative Comparison Across Variants

Figures 18–22 show the same comparison for the 1D Reaction–Diffusion benchmark.

### A.6 1D Wave Equation: Qualitative Comparison Across Variants

Figures 23–27 show the same comparison for the 1D Wave benchmark.

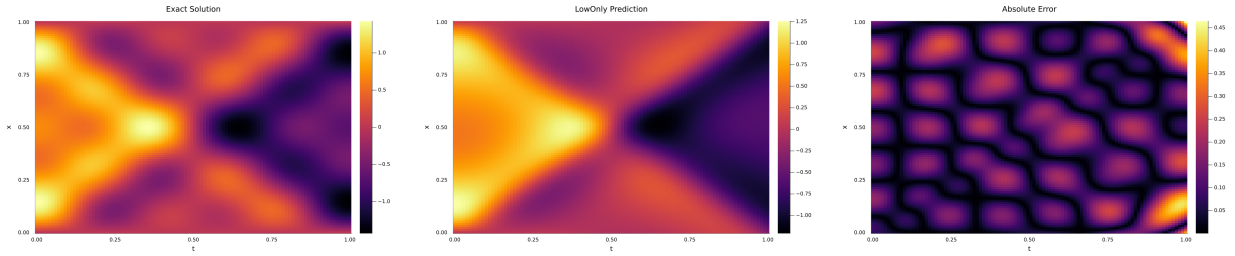



Figure 5: 1D Multimodal Wave, LowOnly variant.

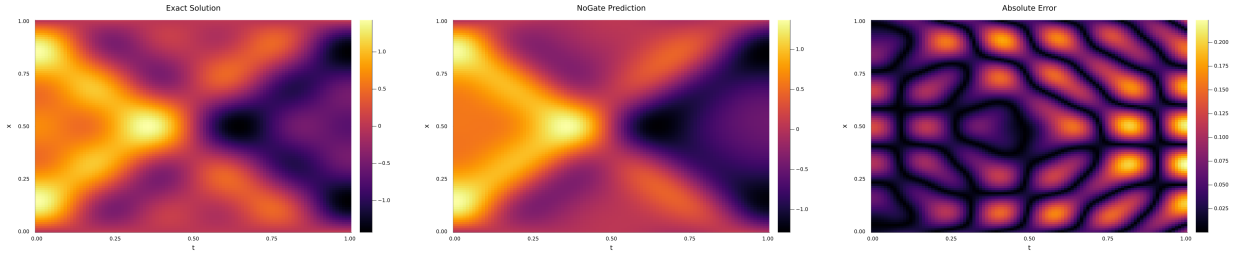
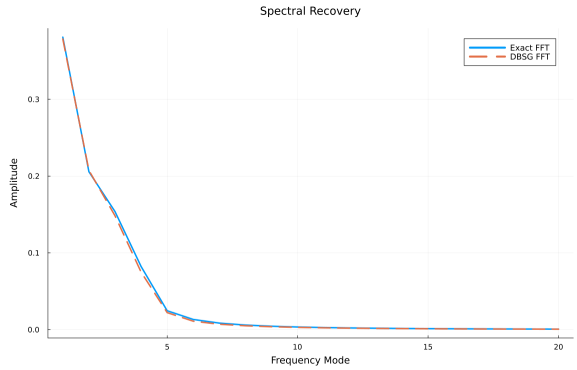
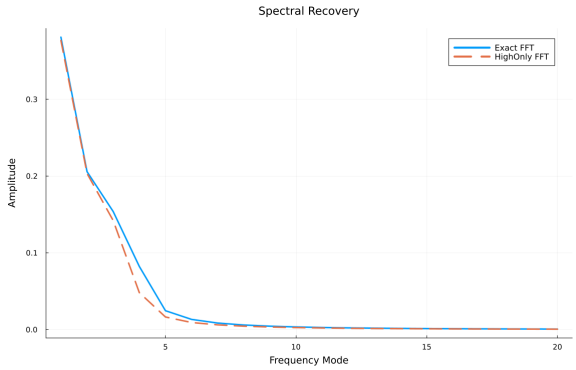


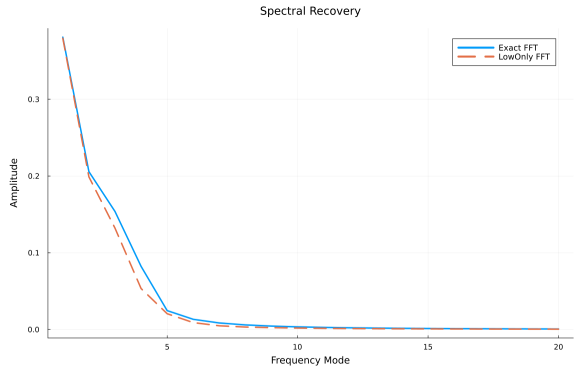

Figure 6: 1D Multimodal Wave, NoGate variant.



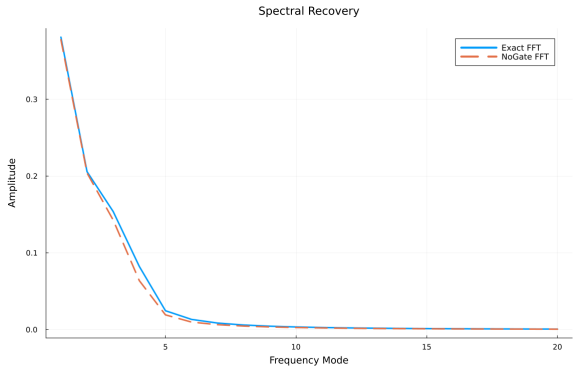

(a) Full (DBSG-PINN)




(b) HighOnly




(c) LowOnly




(d) NoGate

Figure 7: 1D Multimodal Wave spectral recovery across ablation variants (FFT amplitude vs. frequency mode, single seed).

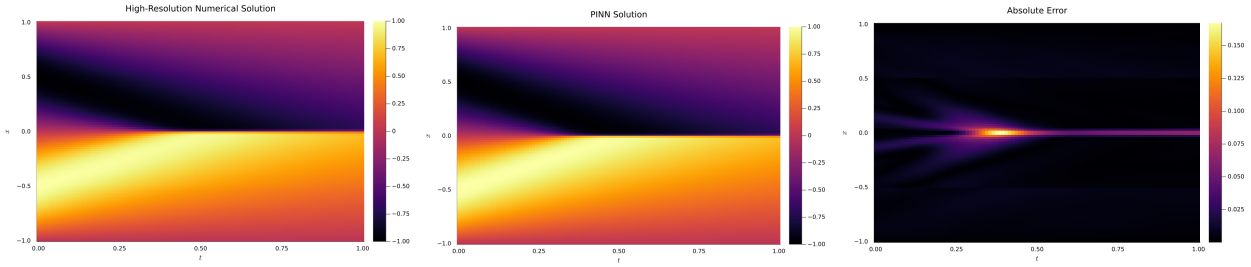



Figure 8: 1D Burgers Equation, Full DBSG-PINN: high-resolution reference solution, prediction, and absolute error.

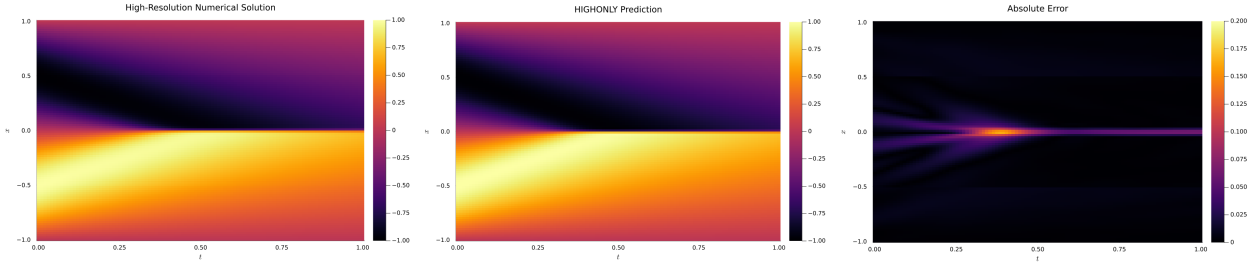



Figure 9: 1D Burgers Equation, HighOnly variant.

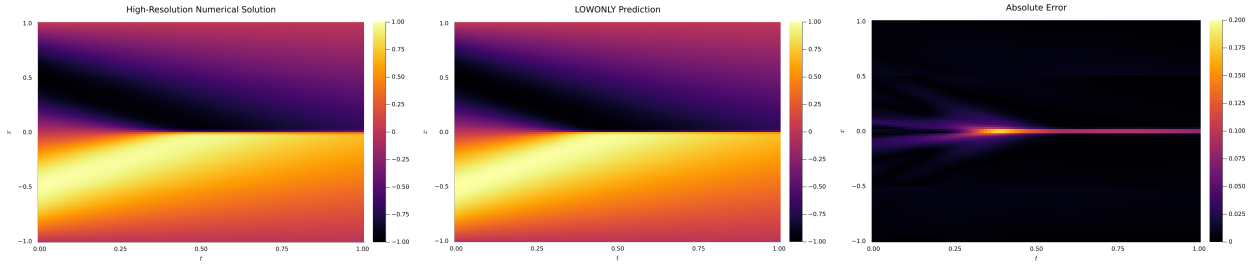



Figure 10: 1D Burgers Equation, LowOnly variant.

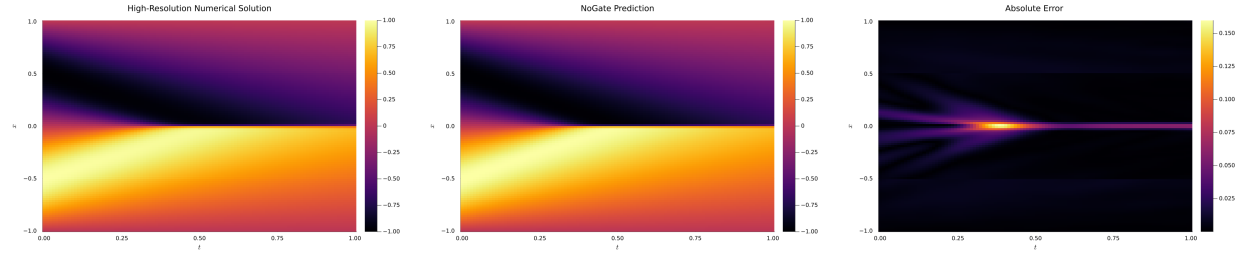
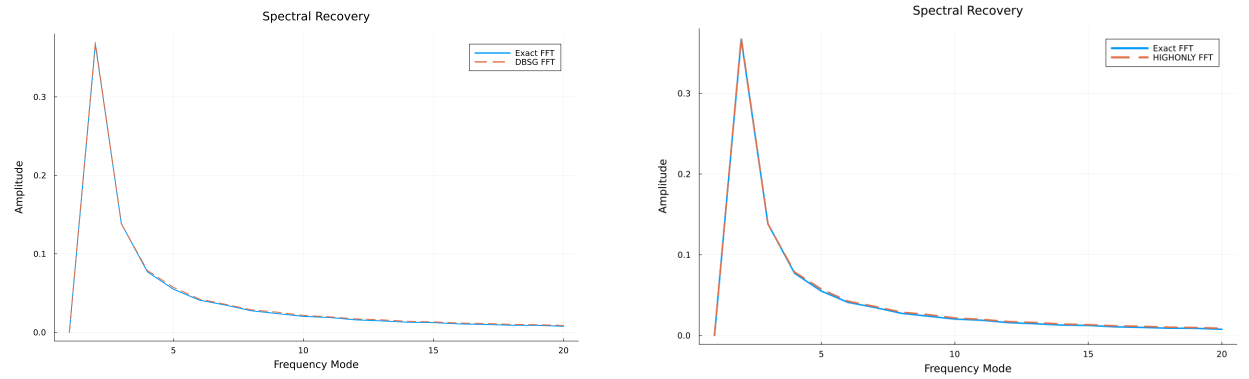


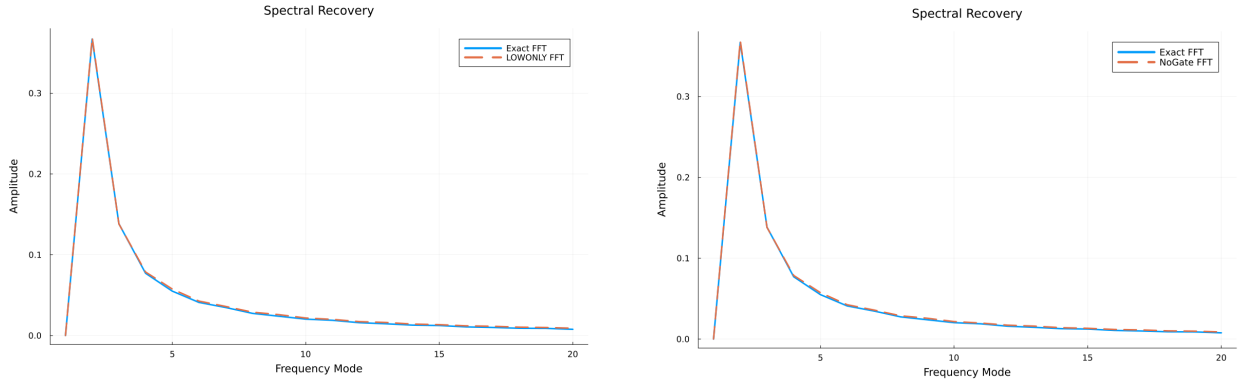

Figure 11: 1D Burgers Equation, NoGate variant.




(a) Full (DBSG-PINN)




(b) HighOnly




(c) LowOnly




(d) NoGate

Figure 12: 1D Burgers Equation spectral recovery across ablation variants (FFT amplitude vs. frequency mode, single seed).

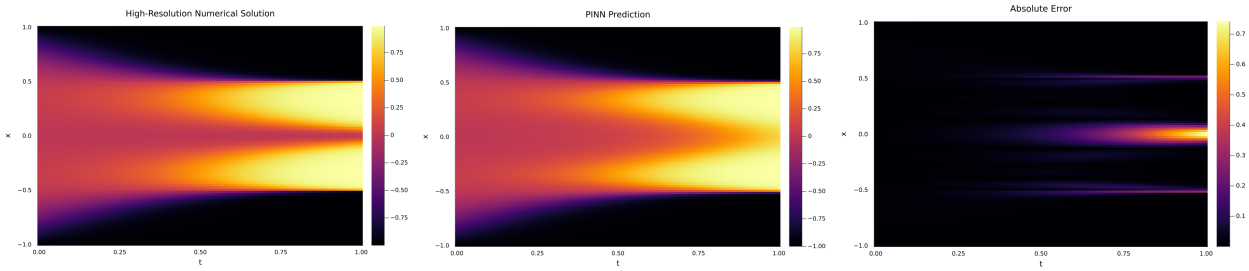



Figure 13: 1D Allen–Cahn Equation, Full DBSG-PINN: high-resolution reference solution, prediction, and absolute error.

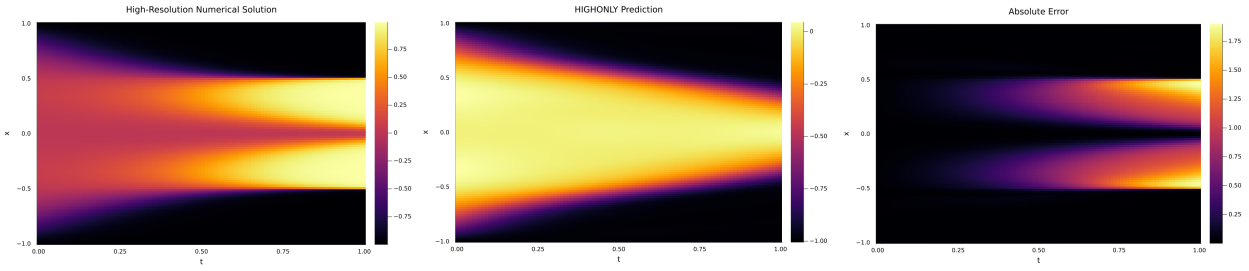



Figure 14: 1D Allen–Cahn Equation, HighOnly variant.

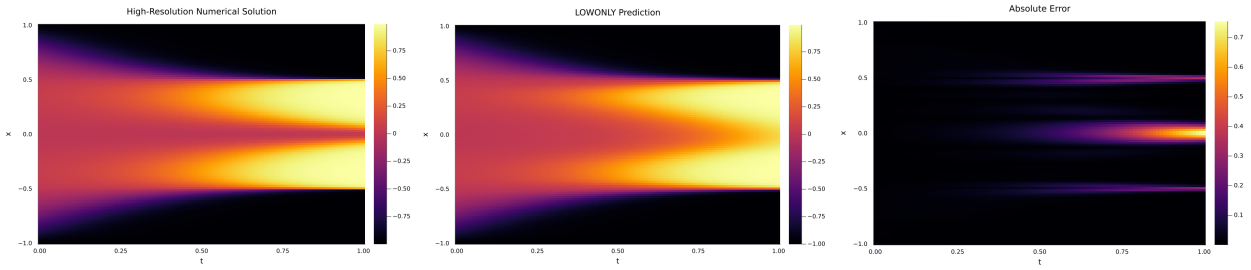



Figure 15: 1D Allen–Cahn Equation, LowOnly variant.

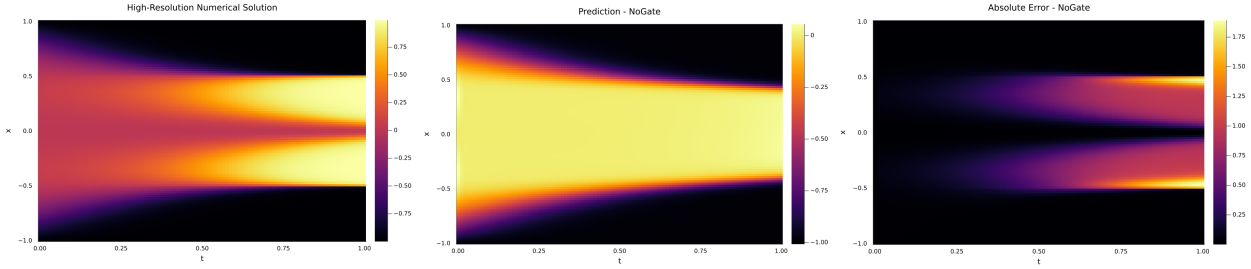
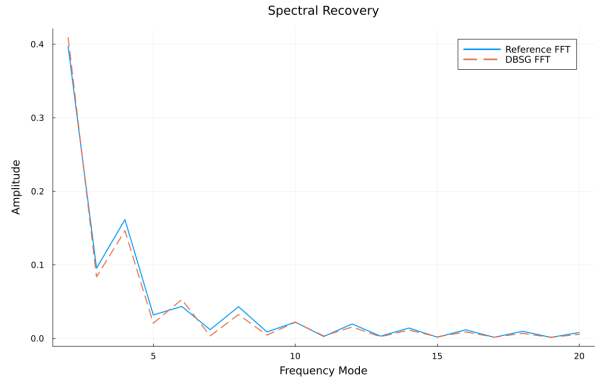
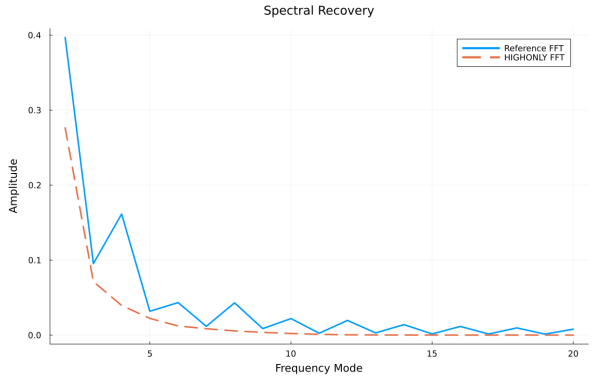


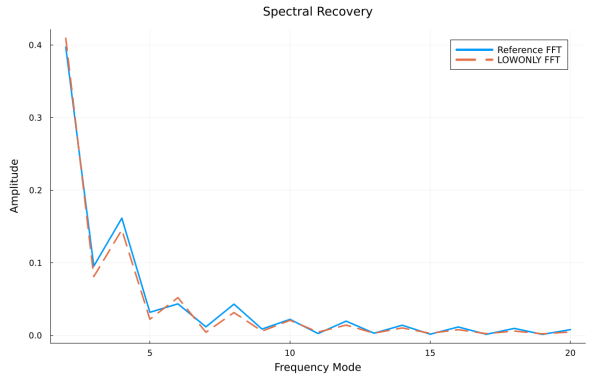

Figure 16: 1D Allen–Cahn Equation, NoGate variant.



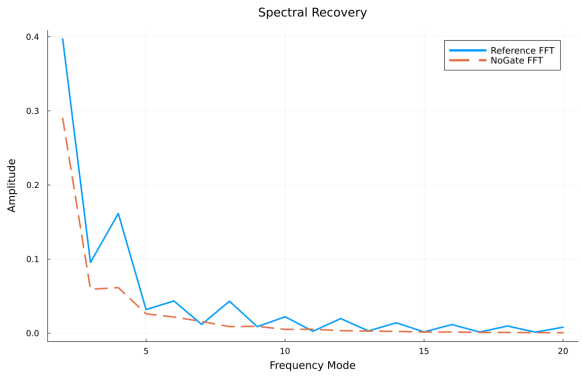

(a) Full (DBSG-PINN)




(b) HighOnly




(c) LowOnly




(d) NoGate

Figure 17: 1D Allen–Cahn Equation spectral recovery across ablation variants (FFT amplitude vs. frequency mode, single seed).

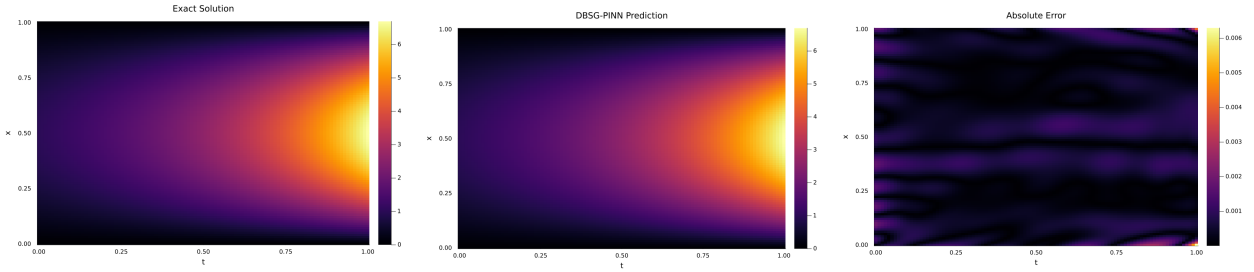



Figure 18: 1D Reaction–Diffusion Equation, Full DBSG-PINN: exact solution, prediction, and absolute error.

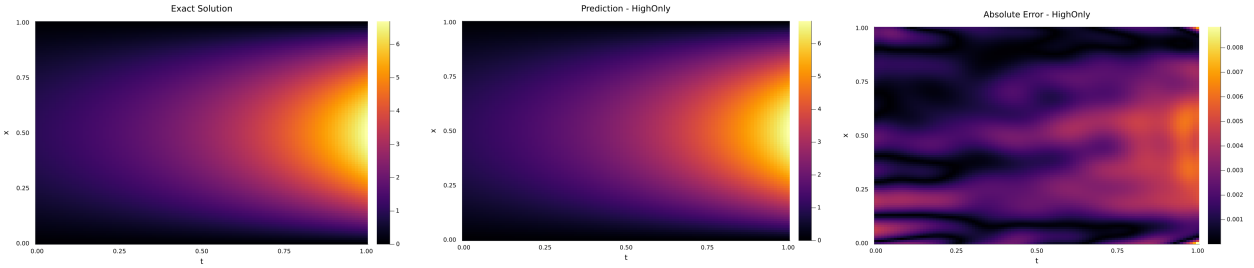



Figure 19: 1D Reaction–Diffusion Equation, HighOnly variant.

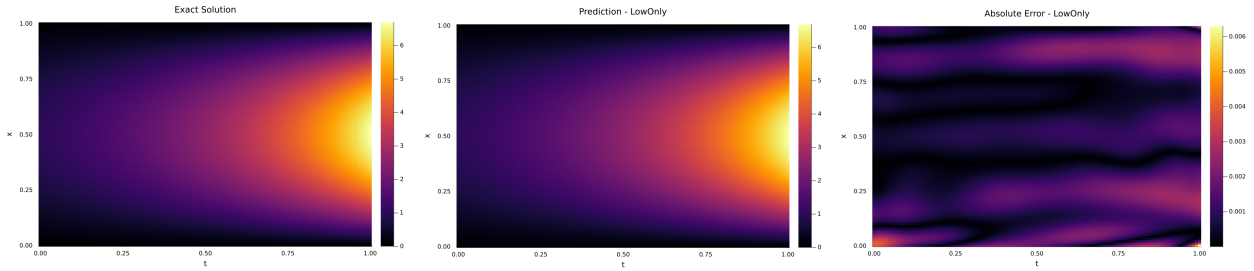



Figure 20: 1D Reaction–Diffusion Equation, LowOnly variant.

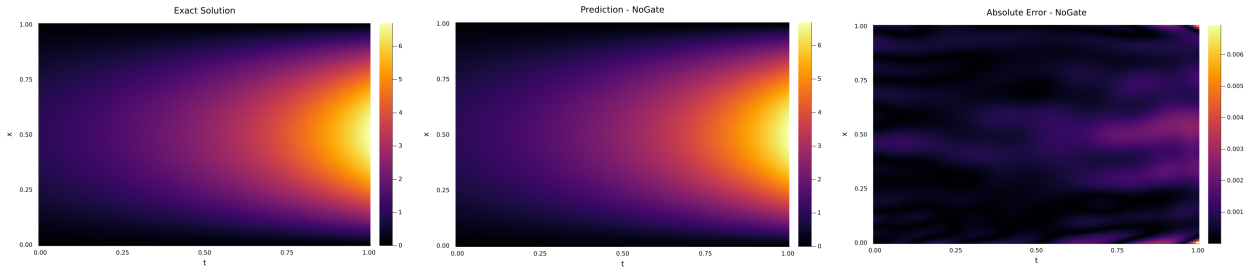
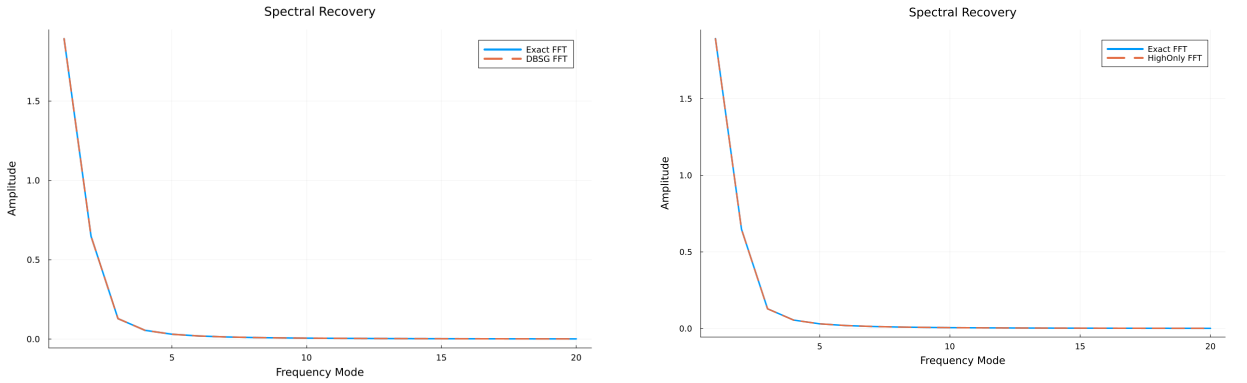
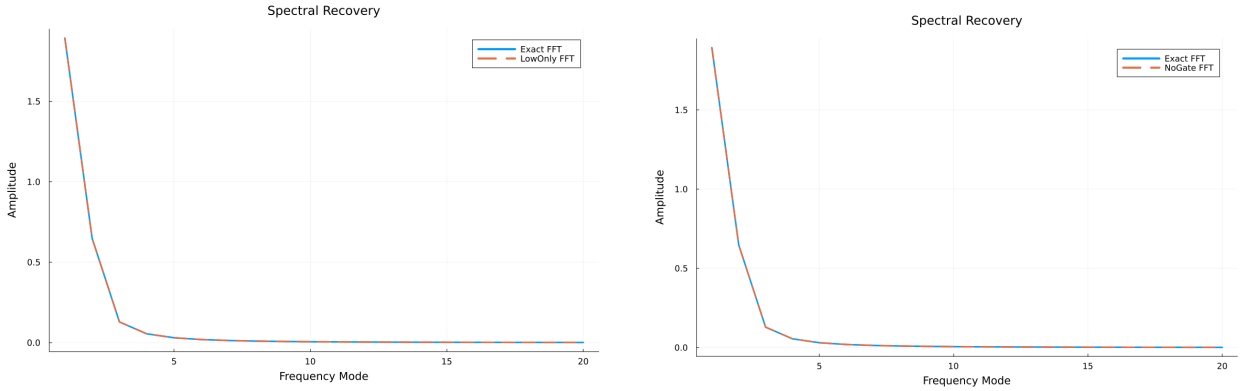



Figure 21: 1D Reaction–Diffusion Equation, NoGate variant.




(a) Full (DBSG-PINN) (b) HighOnly




(c) LowOnly (d) NoGate

Figure 22: 1D Reaction–Diffusion Equation spectral recovery across ablation variants (FFT amplitude vs. frequency mode, single seed).

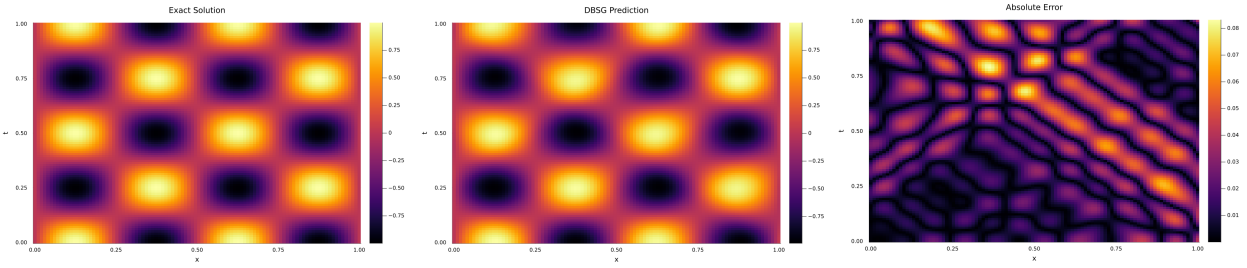



Figure 23: 1D Wave Equation, Full DBSG-PINN: exact solution, prediction, and absolute error.

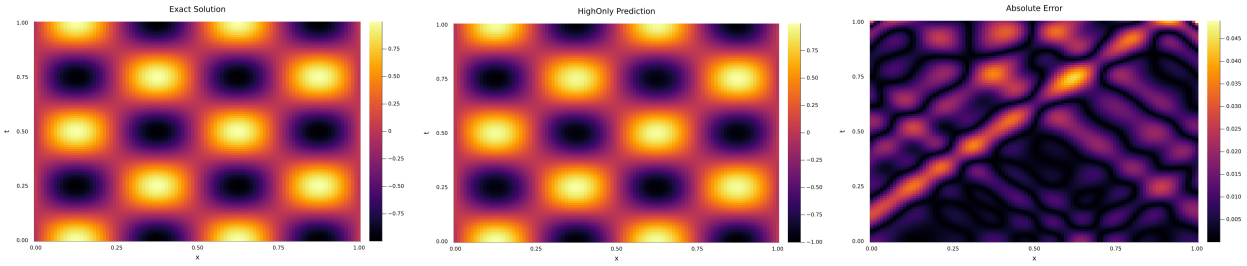



Figure 24: 1D Wave Equation, HighOnly variant.

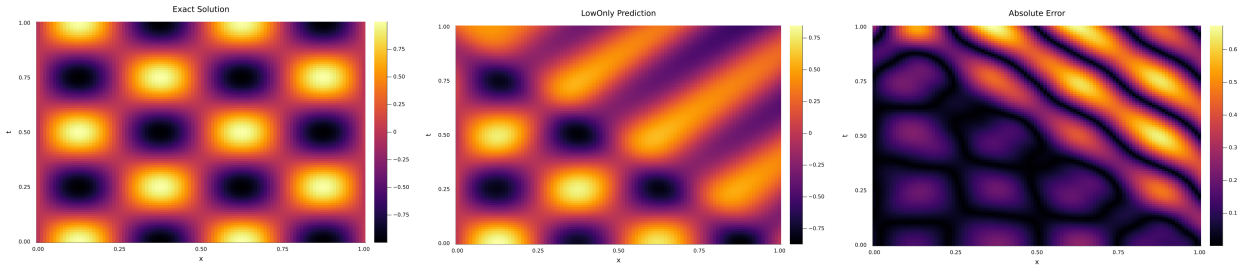



Figure 25: 1D Wave Equation, LowOnly variant.

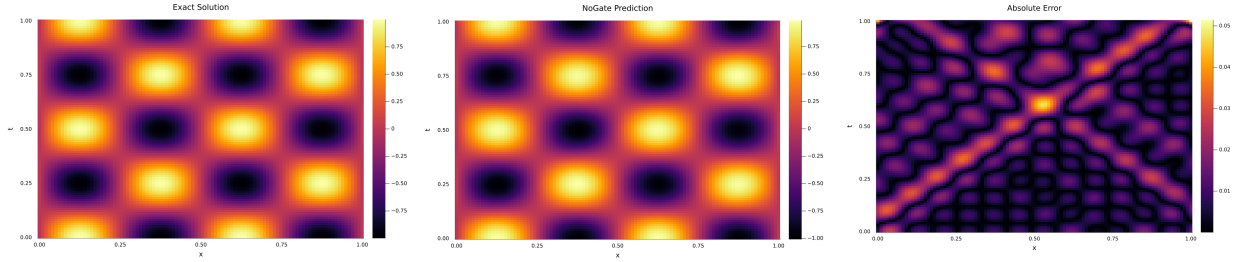
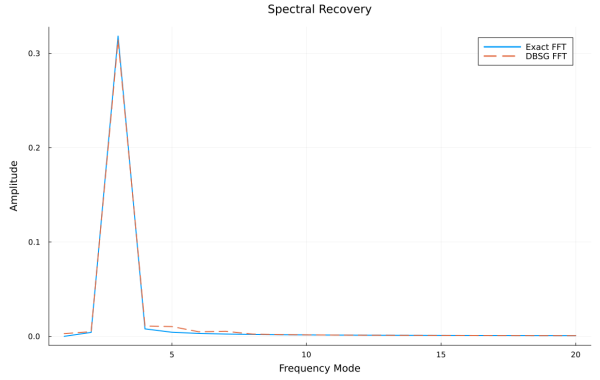
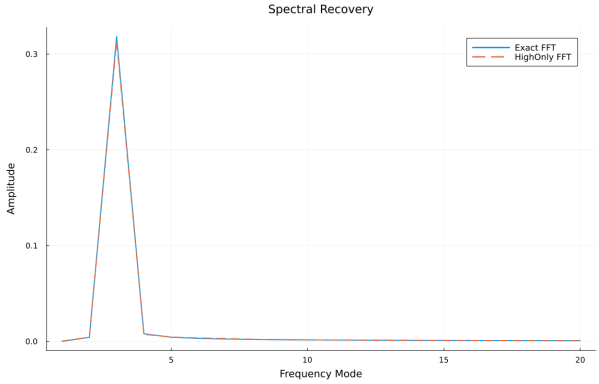


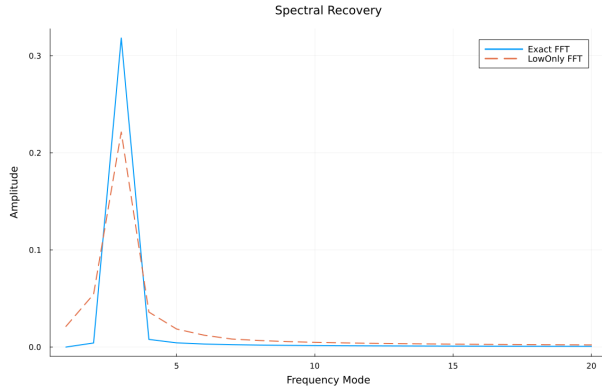

Figure 26: 1D Wave Equation, NoGate variant.



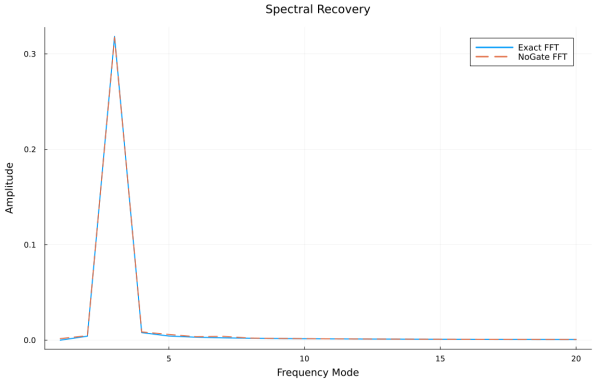

(a) Full (DBSG-PINN)




(b) HighOnly




(c) LowOnly




(d) NoGate

Figure 27: 1D Wave Equation spectral recovery across ablation variants (FFT amplitude vs. frequency mode, single seed).